

APELLIDOS, Nombre:

Nº Matrícula:

---

**U.P.M. E.T.S.I.Inf Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software.**  
*Examen de Programación II. 27 de Marzo de 2014. Primer parcial.*

---

**Realización:** El test se realizará en la hoja de respuesta. Es **importante** rellenar vuestros datos personales y el código clave de vuestro enunciado.

**Duración:** La duración total del test será de **50 minutos**.

**Puntuación:** El test se valora sobre **10 puntos**. Las preguntas tipo test pueden tener una única respuesta o varias respuestas, el enunciado lo deja claro. Cada pregunta con una única respuesta respondida correctamente vale 1 punto, e incorrectamente respondida resta 1/3 puntos. Si en una pregunta con una única respuesta se selecciona más de una respuesta, la pregunta se puntuará con 0 puntos. Para una pregunta con varias respuestas, cada afirmación correcta seleccionada suma 1/*no\_respuestas\_correctas* puntos, y cada afirmación incorrecta seleccionada resta 1/*no\_respuestas\_correctas* puntos. Las preguntas no contestadas suman 0 puntos en cualquier caso.

**Calificaciones:** Las calificaciones se publicarán en moodle como muy tarde el día **31 de Marzo de 2014**

**Revisión:** Las revisiones serán el día **1 de Abril de 2014 de 11:00 a 12:00** previa petición por correo electrónico al profesor Raúl Alonso Calvo, [ralonso@fi.upm.es](mailto:ralonso@fi.upm.es).

---

### Primer Ejercicio

Dado el siguiente fragmento de código:

```
public class Ejercicio1 {
    private Integer [] datos;
    public int nOcupacion;

    public Ejercicio1(int nDatos) {
        // ....
        //Inicialización del array datos para que pueda albergar hasta nDatos
    }

    //Retorna la ocupación
    public int getOcupacion() {
        return this.nOcupacion;
    }

    /*
    * PRE: pos >= 0 && pos < nOcupacion
    * @param pos
    * @return retorna el dato que ocupa la posición indicada
    * si pos no cumple la condición de la PRE retorna null
    */
    public Integer getDatos (int pos) {
        return this.datos[pos];
    }

    //Resto de servicios públicos entre los que se encuentran insertar y
    //borrar
    //...
}
```

### Pregunta 1

Indicar cuál de las siguientes afirmaciones es cierta. (Sólo hay una respuesta correcta.):

- A) El código es correcto porque compila bien y la documentación es válida
  - B) Todas las afirmaciones son incorrectas
  - C) El código es incorrecto porque tiene un atributo público y el método *getDatos* es inconsistente con la documentación
  - D) El código es incorrecto porque tiene un atributo público, pero la documentación es válida
-

## Segundo Ejercicio

Dada la siguiente definición de una clase:

```
public class Ejercicio2 {
    private int dato;
    private int numeroInstancia;
    private static int nInstancias = 0;

    public Ejercicio2 () {
        nInstancias++;
        this.numeroInstancia = nInstancias;
        dato = numeroInstancia * 2;
    }

    /**
     * @return the dato
     */
    public int getDato() {
        return dato;
    }

    /**
     * @return the nInstancias
     */
    public static int getnInstancias() {
        return nInstancias;
    }
}
```

### Pregunta 2

Cuál de las siguientes afirmaciones es cierta. (**Sólo hay una respuesta correcta.**):

- A) No compila porque el constructor debe recibir datos para todos y cada uno de los atributos
- B) No compila porque en el constructor no se ha usado *this* para acceder al valor de *numeroInstancia* en la expresión *dato = numeroInstancia \* 2*;
- C) No compila porque *getDato* accede al atributo privado *dato*
- D) **Todas las afirmaciones son falsas**

## Tercer Ejercicio

Dado el siguiente programa que usa la clase *Ejercicio2* vista en la pregunta anterior:

```
public class PruebaEjercicio2 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        Ejercicio2 ejer2 = new Ejercicio2 ();
        System.out.println("Valor_de_dato:_" +
            Ejercicio2.getDato());
        System.out.println("Número_de_instancias:_" +
            ejer2.getnInstancias());
    }
}
```

### Pregunta 3

Indicar cuál de las siguientes afirmaciones es correcta. (**Sólo hay una respuesta correcta.**):

- A) Ninguna de las afirmaciones es cierta
- B) El código compila perfectamente pero se produce un error de ejecución en la línea *Ejercicio2.getDato()*
- C) El código no compila porque se usa el método de clase *getnInstancias* usando la instancia *ejer2*
- D) **Hay un único error de compilación por usar el método de instancia *getDato* con el nombre de la clase**

**Cuarto Ejercicio**

Dada una clase Fecha que tiene el método público “double diasEntre (Fecha fecha)” que retorna el número de días entre dos fechas, el constructor público Fecha(int anio , int mes, int dia) y el siguiente programa:

```
public static void main(String[] args) {
    int dia=1, mes=1, anio=1970;
    double diasTotales=0;
    Fecha fechas[]=null;
    fechas = new Fecha[20];
    for (int i =0; i< fechas.length; i++){
        fechas[i]= new Fecha (anio+i,mes,dia);
    }

    for (int j =0; j <= fechas.length; j++){
        diasTotales = diasTotales + fechas[j].diasEntre(fechas[j+1]);
    }
    System.out.println("Días totales: "+diasTotales);
}
```

**Pregunta 4**

Suponiendo que el código compila correctamente. Indicar cuál de las afirmaciones es cierta sobre la ejecución de este programa. **(Sólo hay una respuesta correcta.):**

- A) El programa termina de forma anormal debido a un error de ejecución en la línea `diasTotales = diasTotales + fechas[j].diasEntre(fechas[j+1]);`
- B) Todas las afirmaciones son falsas
- C) El código anterior termina mostrando por consola la cadena de caracteres: “Días totales: 0”
- D) No es posible saber cuál será el resultado del programa ya que no se saben los datos de entrada

**Quinto Ejercicio**

Dada la siguiente definición de clases dentro del paquete examen1:

<pre>package examen1;  public class Clase1 {     private int atributo1;     private double atributo2;     // definición de constructores     //...     public double accion1 (int dato){         return atributo2 * dato;     }      double accion2 (double dato){         return atributo2* dato *             accion1 (atributo1);     } }</pre>	<pre>package examen1;  public class Clase2 {     private Clase1 info;     //Definición de constructores     //....     public double calculo (double dato){         return info.accion2(dato);     } }</pre>
--	--

**Pregunta 5**

Indicar cuál de las siguientes afirmaciones es cierta. **(Sólo hay una respuesta correcta.):**

- A) Para que el código compile y funcione correctamente basta con marcar como public el método `accion2`
- B) El código no compila porque el método `accion2` no es accesible desde la Clase2
- C) El código compila y funciona correctamente ya que la Clase1 y la Clase2 están en el mismo paquete
- D) El método `accion2` debe ser marcado como privado ya que es un método que es usado por otro de la misma clase

## Sexto Ejercicio

Dada la implementación de la Clase1 perteneciente al paquete examen1 y la Clase3 que se encuentra en otro paquete distinto:

<pre>package examen1;  public class Clase1 {     private int atributo1;     private double atributo2;     // definición de constructores     //...     public double accion1 (int dato){         return atributo2 * dato;     }      double accion2 (double dato){         return atributo2* dato *             accion1 (atributo1);     } }</pre>	<pre>import examen1.Clase1; public class Clase3 {     private Clase1 info;     //Definición de constructores     //....     public double calculo (double dato){         return info.accion2(dato);     }      public double calculo1 (int dato){         return info.accion1(dato);     } }</pre>
--	--

## Pregunta 6

Suponiendo que la clase Clase1 compila sin errores. Indicar cuál de las siguientes afirmaciones es cierta.

(Sólo hay una respuesta correcta.):

- A) El código de Clase3 compila sin problemas pero el método *calculo* no se ejecutará correctamente
- B) El método *calculo* tiene un error que impide que el código compile ya el método *accion2* no es accesible desde Clase3
- C) El método *calculo1* tiene un error que impide que el código compile ya el método *accion1* no es accesible desde Clase3
- D) El código compila correctamente y se ejecuta sin problemas

## Séptimo Ejercicio

Sean las siguientes afirmaciones sobre constructores.

## Pregunta 7

Indica cuál de las siguientes afirmaciones es cierta. (Sólo hay una respuesta correcta.):

- A) El constructor siempre debe devolver un resultado mediante la sentencia *return*
- B) Es obligatorio definir un método constructor en cualquier clase java, porque java no proporciona uno por defecto
- C) El constructor se ejecuta cuando se crea un objeto mediante la sentencia *new*
- D) Solo puede haber un método constructor en una clase

## Octavo Ejercicio

Dadas las siguientes afirmaciones sobre el atributo “this”

## Pregunta 8

Señalar **todas** las afirmaciones verdaderas. (Puede haber más de una afirmación correcta.):

- A) Se puede usar para evitar colisiones con los identificadores dentro de un método de clase.
- B) Parámetro implícito en todo método de instancia.
- C) Es una referencia al objeto que está ejecutando un método, que permite acceder a todos los atributos y métodos dentro de un método de instancia de la propia clase.
- D) Es una referencia al objeto que está ejecutando un método, y permite acceder a todos sus atributos y métodos dentro de un método de clase.

### Noveno Ejercicio

Dado el siguiente fragmento de código:

```
public class Persona {
    private String nombre;
    private String dni;
    private String fechaNacimiento;

    public Persona(String nombre, String dni, String fechaNacimiento){
        this.nombre=nombre;
        this.dni=dni;
        this.fechaNacimiento=fechaNacimiento;
    }

    /**
     * Devuelve true si las personas comparadas
     * tienen el mismo nombre y fecha de nacimiento
     */
    public boolean equals(Persona otraPersona){
        return nombre.equals(otraPersona.nombre) &&
            fechaNacimiento.equals(otraPersona.fechaNacimiento);
    }

    //Resto de servicios públicos entre los que se encuentran getters y setters
    //...
}
```

#### Pregunta 9

Dado el siguiente programa de prueba:

```
public class PruebaPersona {

    /**
     * @param args
     */
    public static void main(String[] args) {
        Persona juan = new Persona ("Juan_Perez", "54892345S", "12/05/1967");
        Persona juanPerez = new Persona ("Juan_Perez", "00000000A", "12/05/1967");
        Persona juanRodriguez = new Persona ("Juan_Rodriguez", "54892345S", "12/05/1967");
        System.out.print(juan.equals(juanPerez));
        System.out.print(" ");
        System.out.print(juan.equals(juanRodriguez));
        System.out.print(" ");
        System.out.print(juan==juanPerez);
    }
}
```

Indicar la salida de consola. (Sólo hay una respuesta correcta.):

- A) false false false
- B) true false true
- C) true false false
- D) true true true

### Décimo Ejercicio

Dadas las siguientes afirmaciones sobre programación modular:

#### Pregunta 10

Señalar las afirmación verdadera. (Sólo hay una respuesta correcta.):

- A) La parte pública de un módulo recibe el nombre de interfaz
- B) Una librería sólo puede ser utilizada en el programa o proyecto donde se creó
- C) Como norma de buena práctica de programación, los atributos de instancia deben ser siempre públicos
- D) Dos clases pertenecientes a distintos paquetes nunca se pueden llamar igual